

**WEST****End of Result Set**

Generate Collection

Print

L5: Entry 6 of 6

File: USPT

Nov 18, 1997

DOCUMENT-IDENTIFIER: US 5689698 A

\*\* See image for Certificate of Correction \*\*

TITLE: Method and apparatus for managing shared data using a data surrogate and obtaining cost parameters from a data dictionary by evaluating a parse tree object

Detailed Description Text (8):

The client interface 202 provides an interface between the client 220, the federated coordinator 206, and the virtual network 218. The client interface 202 may be resident in the same computer system as the federated coordinator 206, the client 220 or a separate computer, and comprises an open database connectivity module (ODBC) 227 and an object server connectivity module (OSC) 229. In the preferred embodiment, the ODBC module uses MICROSOFT's RTM. Open Database Connectivity technology, which is well known in the art. The ODBC 227 provides an interface between the client 220 and the federated coordinator 206. Since the a command from a client 220 could be either a direct SQL command or a command in another language from an application resident at the client, the ODBC 227 translates object-relational database (ORDB) commands from the client 220 into a form suitable for the federated coordinator 206. In one embodiment, these ORDB commands are translated into Multimedia-SQL (M-SQL), an object relational database language compatible with and derived from SQL. Of course, the actual language implementation is unimportant, and those skilled in the art will recognize that many different languages and protocols can be selected, so long as the ORDB commands are from potentially multiple sources are interpreted and translated into commands that can be understood by the federated coordinator 206. As described herein, the OSC 229 and ODBC 227 are parallel, but not independent, because the ODBC 227 also uses the OSC 229 to redirect object instance data streams to the ODBC 227 control interface to preserve ODBC 227 application interface semantics and to hide the fact that the object data resides on a different data source (such as object server 212) from the RDBMS 210.

Current US Original Classification (1):707/4

**WEST**

Generate Collection

Print

L14: Entry 3 of 7

File: USPT

May 29, 2001

DOCUMENT-IDENTIFIER: US 6240417 B1

TITLE: Integration of legacy database management systems with  
ODBC-compliant application programsAbstract Text (1):

A database integrator is provided to an application program that utilizes the ODBC interface so as to enable the application program to access the data in a legacy DBMS. The application program ("the client program") is located on a workstation, and the legacy DBMS is located on a host computer. The legacy DBMS is used primarily by an application program ("the server program"). The database integrator receives ODBC commands from the client program, and unlike database drivers interacting with non-legacy DBMSs that send SQL statements to the DBMS, the database integrator instead converts these commands into server program-specific commands to manipulate the user interface of the server program, establishes a connection with the server program over a terminal emulation session, and issues the server program-specific commands to the server program by using the terminal emulator. These server program-specific commands manipulate the server program's user interface to display the appropriate data so that the database integrator can then access it. In this manner, the database integrator accesses the data in the legacy DBMS indirectly by issuing user interface commands to the server program, because the database integrator cannot typically access the legacy DBMS directly. This indirect access enables the client program to gain access to the data in a database that it would not otherwise be able to access.

Brief Summary Text (7):

A database integrator is provided to an application program that utilizes the ODBC interface so as to enable the application program to access the data in a legacy DBMS. The application program ("the client program") is located on a workstation, and the legacy DBMS is located on a host computer. The legacy DBMS is used primarily by an application program ("the server program"). The database integrator receives ODBC commands from the client program, and unlike database drivers interacting with non-legacy DBMSs that send SQL statements to the DBMS, the database integrator instead converts these commands into server program-specific commands to manipulate the user interface of the server program, connects to the server program over a terminal emulation session, and issues the server program-specific commands to the server program by using the terminal emulator. These server program-specific commands manipulate the server program's user interface to display the

**WEST**

Generate Collection

Print

L14: Entry 5 of 7

File: USPT

Apr 18, 2000

DOCUMENT-IDENTIFIER: US 6052685 A

TITLE: Integration of legacy database management systems with  
ODBC-compliant application programsAbstract Text (1):

A database integrator is provided to an application program that utilizes the ODBC interface so as to enable the application program to access the data in a legacy DBMS. The application program ("the client program") is located on a workstation, and the legacy DBMS is located on a host computer. The legacy DBMS is used primarily by an application program ("the server program"). The database integrator receives ODBC commands from the client program, and unlike database drivers interacting with non-legacy DBMSs that send SQL statements to the DBMS, the database integrator instead converts these commands into server program-specific commands to manipulate the user interface of the server program, establishes a connection with the server program over a terminal emulation session, and issues the server program-specific commands to the server program by using the terminal emulator. These server program-specific commands manipulate the server program's user interface to display the appropriate data so that the database integrator can then access it. In this manner, the database integrator accesses the data in the legacy DBMS indirectly by issuing user interface commands to the server program, because the database integrator cannot typically access the legacy DBMS directly. This indirect access enables the client program to gain access to the data in a database that it would not otherwise be able to access.

Brief Summary Text (7):

A database integrator is provided to an application program that utilizes the ODBC interface so as to enable the application program to access the data in a legacy DBMS. The application program ("the client program") is located on a workstation, and the legacy DBMS is located on a host computer. The legacy DBMS is used primarily by an application program ("the server program"). The database integrator receives ODBC commands from the client program, and unlike database drivers interacting with non-legacy DBMSs that send SQL statements to the DBMS, the database integrator instead converts these commands into server program-specific commands to manipulate the user interface of the server program, connects to the server program over a terminal emulation session, and issues the server program-specific commands to the server program by using the terminal emulator. These server program-specific commands manipulate the server program's user interface to display the

**WEST**

Generate Collection

Print

L2: Entry 9 of 17

File: DWPI

Apr 17, 2001

DERWENT-ACC-NO: 2001-482087

DERWENT-WEEK: 200152

COPYRIGHT 2003 DERWENT INFORMATION LTD

TITLE: Network resource management system, has network module with options database to generate hypertext transfer protocol requests by user client to be processed by web server

Basic Abstract Text (2):

DETAILED DESCRIPTION - The system has a network resource including a network module (400) which interfaces the resource to the network. The module has an options database (430) including selectable options for operations relating to the network resource. The operations include reporting, configuration and control of the network resource. A native resource instructions library (410) includes native resource instructions for the resource. A instruction module (420) translates function calls into native resource instructions for the resource using the library (410). A web server transfers the selectable options over the network using the hypertext transfer protocol. The web server processes hypertext transfer protocol requests from the client including one or more hyperlink references. An INDEPENDENT CLAIM is also included for method for remote control of hardware in the network device connected to a network by client.

**WEST****End of Result Set**

Generate Collection

Print

L2: Entry 17 of 17

File: DWPI

Oct 16, 1997

DERWENT-ACC-NO: 1997-512941  
DERWENT-WEEK: 200035  
COPYRIGHT 2003 DERWENT INFORMATION LTD

TITLE: Schema integration method for distributed heterogeneous databases - involves defining equivalence of objects of normalised schemas which are integrated to produce global object oriented schema which are then converted into relational form

Basic Abstract Text (2):

The equivalence of objects of the two normalised schemes are defined and integrated to produce global object oriented schema. The schema are then converted into relational from SQL commands are developed to realise the global schema within the DDA as a virtual database system satisfying all the requirements for accessing data from the independent database systems.

Equivalent Abstract Text (2):

The equivalence of objects of the two normalised schemes are defined and integrated to produce global object oriented schema. The schema are then converted into relational from SQL commands are developed to realise the global schema within the DDA as a virtual database system satisfying all the requirements for accessing data from the independent database systems.

**WEST**

Generate Collection

Print

L2: Entry 8 of 17

File: DWPI

Nov 6, 2002

DERWENT-ACC-NO: 2001-425088  
DERWENT-WEEK: 200316  
COPYRIGHT 2003 DERWENT INFORMATION LTD

TITLE: Data transmission system and sale managing system

INVENTOR: YAMADA, T

PATENT-ASSIGNEE: VISUAL JAPAN KK (VISUN)

PRIORITY-DATA: 2000JP-0308237 (October 6, 2000), 1999JP-0291162  
(October 13, 1999)

## PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
CN 1378670 A	November 6, 2002		000	G06F015/00
WO 200127778 A1	April 19, 2001	J	063	G06F015/00
JP 2001184307 A	July 6, 2001		022	G06F015/00
EP 1223517 A1	July 17, 2002	E	000	G06F015/00
KR 2002038820 A	May 23, 2002		000	G06F015/00

DESIGNATED-STATES: CN KR US AT BE CH CY DE DK ES FI FR GB GR IE IT  
LU MC NL PT SE AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL  
PT SE

## APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
CN 1378670A	October 12, 2000	2000CN-0814206	
WO 200127778A1	October 12, 2000	2000WO-JP07064	
JP2001184307A	October 6, 2000	2000JP-0308237	
EP 1223517A1	October 12, 2000	2000EP-0966423	
EP 1223517A1	October 12, 2000	2000WO-JP07064	
EP 1223517A1		WO 200127778	Based on
KR2002038820A	April 13, 2002	2002KR-0704769	

INT-CL (IPC): G06 F 12/00; G06 F 13/00; G06 F 15/00; G06 F 17/30;  
G06 F 17/60

ABSTRACTED-PUB-NO: WO 200127778A  
BASIC-ABSTRACT:

NOVELTY - A network data transmission system where a server (4(5))

(19) 世界知的所有権機関  
国際事務局



(43) 国際公開日  
2001年4月19日 (19.04.2001)

PCT

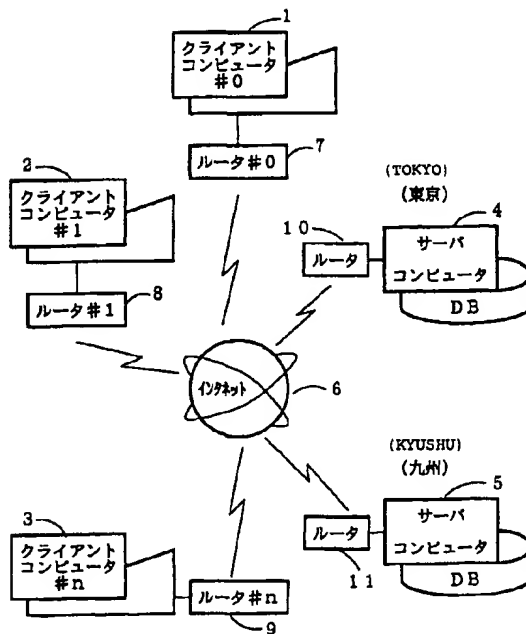
(10) 国際公開番号  
WO 01/27778 A1

- (51) 国際特許分類<sup>7</sup>: G06F 15/00, 特願平 2000-308237  
17/30, 17/60, 12/00, 13/00 2000年10月6日 (06.10.2000) JP
- (21) 国際出願番号: PCT/JP00/07064 (71) 出願人 (米国を除く全ての指定国について): 株式会社  
ビジュアルジャパン (KABUSHIKI KAISHA VISUAL  
(22) 国際出願日: 2000年10月12日 (12.10.2000) JAPAN) [JP/IP]; 〒101-0051 東京都千代田区神田神保  
(25) 国際出願の言語: 日本語 町3-12-3 Tokyo (JP).
- (26) 国際公開の言語: 日本語 (72) 発明者; および  
(75) 発明者/出願人 (米国についてのみ): 山田徳廣 (YA-  
(30) 優先権データ: 特願平 11/291162 MADA, Tokuhiko) [JP/IP]; 〒101-0051 東京都千代田区  
1999年10月13日 (13.10.1999) JP 神田神保町3-12-3 株式会社 ビジュアルジャパン内  
Tokyo (JP).

[続葉有]

(54) Title: DATA TRANSMISSION SYSTEM AND SALE MANAGING SYSTEM

(54) 発明の名称: ネットワーク型データ伝送方法ならびにシステムおよび同システムにおけるサーバあるいは端末装置、ならびに同方法がプログラムされ記録された記録媒体および同方法を用いた販売管理システム



- 1...CLIENT COMPUTER #0  
2...CLIENT COMPUTER #1  
3...CLIENT COMPUTER #n  
4...SERVER COMPUTER  
5...SERVER COMPUTER  
6...THE INTERNET  
7...ROUTER #0  
8...ROUTER #1  
9...ROUTER #n  
10...ROUTER  
11...ROUTER

(57) Abstract: A network data transmission system where a server (4(5)) and terminals (client computers) (1(2, 3)) sharing the server (4(5)) are interconnected through a network circuit (6), wherein each time a terminal collects data, packet creating means (223) incorporated in the terminal issues a packet command complying with DB command transmission protocols independent of the network circuit, and packet converting means (423) that the server receiving the packet command has converts the packet command to a desired DB operation command and executes it, thus allowing the DB to reflect the contents and responding the request from the terminal in real time. By mounting only a communication driver realizing TCP-IP, the terminal can operate the database server, and an effective sale managing system such as a POS can be structured using this.

[続葉有]

**WEST**

Generate Collection

Print

L10: Entry 9 of 22

File: USPT

Sep 24, 2002

DOCUMENT-IDENTIFIER: US 6457003 B1

TITLE: Methods, systems and computer program products for logical access of data sources utilizing standard relational database management systems

Detailed Description Text (8):

As shown in FIG. 2, the memory 36 may include several major categories of software and data used in the computer system 30: the operating system 52; the application programs 54; the ODBC driver 50; the logical translation module 60; the input/output (I/O) device drivers 58; and the data 56. The I/O device drivers 58 typically include software routines accessed through the operating system 52 by the application programs 54 to communicate with devices such as the input devices 32, the display 34, the speaker 44, the microphone 42, the I/O data port(s) 46, and certain memory 36 components. The application programs 54 comprise the programs that implement the various features of the computer system 30 and preferably include at least one application module or object for accessing data sources utilizing the ODBC driver 50. The ODBC driver 50 is preferably part of a standard relational database management system which provides access to relational data sources and preferably includes the ability to exit the ODBC driver 50 to provide relational database commands to the logical translation module 60 for translation. More preferably, the ODBC driver 50 is associated with a relational database management system such as International Business Machine Corporation's DataJoiner which provides access to both relational and non-relational data sources. Alternatively, the logical translation module 60 could intercept commands intended for the ODBC driver 50, perform the logical to physical translation as described below, and then provide the translated commands to the ODBC driver 50.

Detailed Description Text (12):

As is illustrated in FIG. 3, an application program 54 provides an ODBC command to the ODBC driver 50. The ODBC driver 50 preferably provides an interface for accessing a standard database management system such as International Business Machine Corporation's DB2 DataJoiner product. Thus, an application which accesses DataJoiner may do so by accessing the ODBC interface of DataJoiner. In any event, the ODBC driver 50 provides the received command to the logical to physical translation module 60. The logical to physical translation module 60 examines the logical information in the command and accesses the logical to physical mapping metadata tables (i.e. the translation tables) 70. Tables 70 contain metadata to translate the command which includes logical information, such



**WEST**

Generate Collection

Print

L10: Entry 12 of 22

File: USPT

Nov 27, 2001

DOCUMENT-IDENTIFIER: US 6324541 B1

TITLE: System, method, and computer program product for providing relational patterns between entities

Detailed Description Text (68):

In an embodiment of the present invention, the software code communicates with the databases 102 and 104 using the Open Database Connectivity (ODBC) interface. As is well known in the relevant art(s), ODBC is a standard for accessing different database systems from a high level programming language application. It enables these applications to submit statements to ODBC using an ODBC structured query language (SQL) and then translates these to the particular SQL commands the underlying database product employs. In an alternate embodiment, the CIDM processing software communicates with the databases 102 and 104 using the Active Data Objects (ADO) interface.

Current US Cross Reference Classification (4):707/4

**WEST**

Generate Collection

Print

L10: Entry 2 of 22

File: PGPB

Mar 7, 2002

DOCUMENT-IDENTIFIER: US 20020029212 A1

TITLE: Method, system, and program for using a fetch request to make data available to an application program

Current US Classification, US Primary Class/Subclass (1):  
707/3Summary of Invention Paragraph (8):

[0007] Open DataBase Connectivity (ODBC) is a standard database access method to allow applications to access data in a database management system (DBMS). An ODBC driver translates the application's queries into commands that the DBMS understands. The ODBC standards describe scrollable, keyset driven, static and dynamic cursors. The ODBC standards mention that cursors may be updateable or nonupdateable. Cursors are defined as updateable if the application is capable of modifying the data in the cursor result table. As discussed, the result table may be implemented in a work file or comprise the rows pointed to by the cursor in the base table. The ODBC also mentions that when positioned on a row in an updateable cursor, the application can perform position updates or delete operations that target the base table rows used to build the current row in the cursor.

**WEST**

Generate Collection

Print

L10: Entry 1 of 22

File: PGPB

Mar 14, 2002

DOCUMENT-IDENTIFIER: US 20020032678 A1

TITLE: Method, system, and program for processing a fetch request for a target row at an absolute position from a first entry in a table

Current US Classification, US Primary Class/Subclass (1):  
707/3

Summary of Invention Paragraph (8):

[0007] Open DataBase Connectivity (ODBC) is a standard database access method to allow applications to access data in a database management system (DBMS). An ODBC driver translates the application's queries into commands that the DBMS understands. The ODBC standards describe scrollable, keyset driven, static and dynamic cursors. The ODBC standards mention that cursors may be updateable or nonupdateable. Cursors are defined as updateable if the application is capable of modifying the data in the cursor result table. As discussed, the result table may be implemented in a work file or comprise the rows pointed to by the cursor in the base table. The ODBC also mentions that when positioned on a row in an updateable cursor, the application can perform position updates or delete operations that target the base table rows used to build the current row in the cursor.

**WEST**

Generate Collection

Print

L5: Entry 2 of 6

File: USPT

Feb 16, 1999

DOCUMENT-IDENTIFIER: US 5873083 A

TITLE: Method and apparatus for extending a relational database management system using a federated coordinator

Detailed Description Text (8):

The client interface 202 provides an interface between the client 220, the federated coordinator 206, and the virtual network 218. The client interface 202 may be resident in the same computer system as the federated coordinator 206, the client 220 or a separate computer, and comprises an open database connectivity module (ODBC) 227 and an object server connectivity module (OSC) 229. In the preferred embodiment, the ODBC module uses MICROSOFT's RTM. Open Database Connectivity technology, which is well known in the art. The ODBC 227 provides an interface between the client 220 and the federated coordinator 206. Since the a command from a client 220 could be either a direct SQL command or a command in another language from an application resident at the client, the ODBC 227 translates object-relational database (ORDB) commands from the client 220 into a form suitable for the federated coordinator 206. In one embodiment, these ORDB commands are translated into Multimedia-SQL (M-SQL), an object relational database language compatible with and derived from SQL. Of course, the actual language implementation is unimportant, and those skilled in the art will recognize that many different languages and protocols can be selected, so long as the ORDB commands are from potentially multiple sources are interpreted and translated into commands that can be understood by the federated coordinator 206. As described herein, the OSC 229 and ODBC 227 are parallel, but not independent, because the ODBC 227 also uses the OSC 229 to redirect object instance data streams to the ODBC 227 control interface to preserve ODBC 227 application interface semantics and to hide the fact that the object data resides on a different data source (such as object server 212) from the RDBMS 210.

Current US Original Classification (1):

707/4

Current US Cross Reference Classification (2):

707/3

**WEST**

Generate Collection

Print

L5: Entry 5 of 6

File: USPT

May 19, 1998

DOCUMENT-IDENTIFIER: US 5754841 A

TITLE: Method and apparatus for parallel execution of user-defined functions in an object-relational database management system

Detailed Description Text (8):

The client interface 202 provides an interface between the client 220, the federated coordinator 206, and the virtual network 218. The client interface 202 may be resident in the same computer system as the federated coordinator 206, the client 220 or a separate computer, and comprises an open database connectivity module (ODBC) 227 and an object server connectivity module (OSC) 229. In the preferred embodiment, the ODBC module uses MICROSOFT's RTM. Open Database Connectivity technology, which is well known in the art. The ODBC 227 provides an interface between the client 220 and the federated coordinator 206. Since the a command from a client 220 could be either a direct SQL command or a command in another language from an application resident at the client, the ODBC 227 translates object-relational database (ORDB) commands from the client 220 into a form suitable for the federated coordinator 206. In one embodiment, these ORDB commands are translated into Multimedia-SQL (M-SQL), an object relational database language compatible with and derived from SQL. Of course, the actual language implementation is unimportant, and those skilled in the art will recognize that many different languages and protocols can be selected, so long as the ORDB commands are from potentially multiple sources are interpreted and translated into commands that can be understood by the federated coordinator 206. As described herein, the OSC 229 and ODBC 227 are parallel, but not independent, because the ODBC 227 also uses the OSC 229 to redirect object instance data streams to the ODBC 227 control interface to preserve ODBC 227 application interface semantics and to hide the fact that the object data resides on a different data source (such as object server 212) from the RDBMS 210.

Current US Original Classification (1):

707/3

**WEST**

Generate Collection

Print

L10: Entry 9 of 22

File: USPT

Sep 24, 2002

DOCUMENT-IDENTIFIER: US 6457003 B1

TITLE: Methods, systems and computer program products for logical access of data sources utilizing standard relational database management systems

Detailed Description Text (8):

As shown in FIG. 2, the memory 36 may include several major categories of software and data used in the computer system 30: the operating system 52; the application programs 54; the ODBC driver 50; the logical translation module 60; the input/output (I/O) device drivers 58; and the data 56. The I/O device drivers 58 typically include software routines accessed through the operating system 52 by the application programs 54 to communicate with devices such as the input devices 32, the display 34, the speaker 44, the microphone 42, the I/O data port(s) 46, and certain memory 36 components. The application programs 54 comprise the programs that implement the various features of the computer system 30 and preferably include at least one application module or object for accessing data sources utilizing the ODBC driver 50. The ODBC driver 50 is preferably part of a standard relational database management system which provides access to relational data sources and preferably includes the ability to exit the ODBC driver 50 to provide relational database commands to the logical translation module 60 for translation. More preferably, the ODBC driver 50 is associated with a relational database management system such as International Business Machine Corporation's DataJoiner which provides access to both relational and non-relational data sources. Alternatively, the logical translation module 60 could intercept commands intended for the ODBC driver 50, perform the logical to physical translation as described below, and then provide the translated commands to the ODBC driver 50.

Detailed Description Text (12):

As is illustrated in FIG. 3, an application program 54 provides an ODBC command to the ODBC driver 50. The ODBC driver 50 preferably provides an interface for accessing a standard database management system such as International Business Machine Corporation's DB2 DataJoiner product. Thus, an application which accesses DataJoiner may do so by accessing the ODBC interface of DataJoiner. In any event, the ODBC driver 50 provides the received command to the logical to physical translation module 60. The logical to physical translation module 60 examines the logical information in the command and accesses the logical to physical mapping metadata tables (i.e. the translation tables) 70. Tables 70 contain metadata to translate the command which includes logical information, such

## WEST Search History

DATE: Thursday, October 30, 2003

**Set Name Query**  
**side by side**

**Hit Count Set Name**  
**result set**

*DB=USPT; PLUR=YES; OP=ADJ*

L6	l3 same (open\$ or independent\$)	9	L6
L5	l3 and l4	6	L5
L4	(707/4 OR 707/1 OR 707/3).CCLS.	3907	L4
L3	L2 same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	33	L3
L2	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) near12 protocol\$	410	L2
L1	5930786[pn]	1	L1

END OF SEARCH HISTORY

# WEST Search History

DATE: Thursday, October 30, 2003

## Set Name Query

side by side

## Hit Count Set Name

result set

*DB=USPT,PGPB; PLUR=YES; OP=ADJ*

L10	l6 and l7	22	L10
L9	l6[ti,ab]	2	L9
L8	l6 and l7	22	L8
L7	(707/1 OR 707/3 OR 707/4 OR 709/230 OR 709/250 OR 709/213 OR 709/216 OR 709/217 OR 711/100).CCLS.	11549	L7
L6	l5 same (odbc or (open adj (database or (data base)) adj connect\$))	45	L6
L5	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	1354	L5

*DB=EPAB,DWPI; PLUR=YES; OP=ADJ*

L4	L3 and l1	2	L4
L3	odbc or (open adj (database or (data base)) adj connect\$)	42	L3
L2	L1 same (open or independent)	17	L2
L1	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	149	L1

END OF SEARCH HISTORY



# WEST Search History

DATE: Thursday, October 30, 2003

<u>Set Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
side by side			result set
<i>DB=USPT,PGPB; PLUR=YES; OP=ADJ</i>			
L9	l6[ti,ab]	2	L9
L8	l6 and l7	22	L8
L7	(707/1 OR 707/3 OR 707/4 OR 709/230 OR 709/250 OR 709/213 OR 709/216 OR 709/217 OR 711/100).CCLS.	11549	L7
L6	l5 same (odbc or (open adj (database or (data base)) adj connect\$))	45	L6
L5	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	1354	L5
<i>DB=EPAB,DWPI; PLUR=YES; OP=ADJ</i>			
L4	L3 and l1	2	L4
L3	odbc or (open adj (database or (data base)) adj connect\$)	42	L3
L2	L1 same (open or independent)	17	L2
L1	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	149	L1

END OF SEARCH HISTORY

## WEST Search History

DATE: Thursday, October 30, 2003

Set Name Query  
side by side

Hit Count Set Name  
result set

*DB=EPAB,DWPI; PLUR=YES; OP=ADJ*

L4	L3 and l1	2	L4
L3	odbc or (open adj (database or (data base)) adj connect\$)	42	L3
L2	L1 same (open or independent)	17	L2
L1	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	149	L1

END OF SEARCH HISTORY

# WEST Search History

DATE: Thursday, October 30, 2003

## Set Name Query side by side

## Hit Count Set Name result set

*DB=USPT; PLUR=YES; OP=ADJ*

L14	l11 and L13	7	L14
L13	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	945	L13
L12	L11 and l2	3	L12
L11	l7 or l8 or l9 or l10	68	L11
L10	5873083[uref]	17	L10
L9	5689698[uref]	19	L9
L8	5754841[uref]	34	L8
L7	5930786[uref]	15	L7
L6	l3 same (open\$ or independent\$)	9	L6
L5	l3 and l4	6	L5
L4	(707/4 OR 707/1 OR 707/3).CCLS.	3907	L4
L3	L2 same (database or (data base) or infobase or (info base) or (information base) or ((stor\$ or memor\$) near2 (pool\$ or block\$)))	33	L3
L2	((command or instruction) near4 (translat\$ or convert\$ or conversion\$ or transform\$)) near12 protocol\$	410	L2
L1	5930786[pn]	1	L1

END OF SEARCH HISTORY